ALGORITMA KLASIFIKASI : K-NEAREST NEIGHBOR (K-NN)

1. Tujuan

1. Mahasiswa mampu menggunakan algoritma k-nearest neighbor (k-nn).
2. Mahasiswa mampu menerapkan algoritma k-nn dalam kasus nyata.

2. Landasan Teori

Algoritma K-Nearest Neighbor (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan k ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruangan ini ditandai dengan kelas c, jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut.

K-NN adalah suatu metode yang menggunakan algoritma supervised dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN. Tujuan dari algoritma ini adalah mengklasifikasi objek baru berdasakan atribut dan sampel latih. pengklasifikasian tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik uji, akan ditemukan sejumlah K objek (titik training) yang paling dekat dengan titik uji. Klasifikasi menggunakan voting terbanyak di antara klasifikasi dari K objek. Algoritma K-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sample uji yang baru. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Eucledian.

Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (klasifikasinya belum diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor training sample dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Ketepatan algoritma K-NN ditentukan oleh ada atau tidak adanya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut setara dengan relevansinya terhadap klasifikasi. Algoritma K-NN memiliki kelebihan yaitu dapat

DATA MINING - WIWIT SP

menghasilkan data yang kuat atau jelas dan efektif jika digunakan pada data yang besar. Dari beberapa kelebihan tersebut, K-NN juga memiliki kekurangan yaitu membutuhkan nilai K sebagai parameter, jarak dari data percobaan dapat tidak jelas dengan tipe jarak yang digunakan dan dengan atribut yang digunakan untuk memperoleh hasil yang terbaik, maka harus menggunakan semua atribut atau hanya satu atribut yang telah pasti, dan perhitungan harga sangat tinggi karena percobaan ini membutuhkan perhitungan jarak dari beberapa query untuk semua data percobaan.

Langkah-langkah dalam metode K-NN adalah sebagai berikut:

1) Tentukan k.
2) Hitung jarak antara data baru ke setiap labeled data.
3) Tentukan k labeled data yang mempunyai jarak yang paling minimal.
4) Klasifikasikan data baru ke dalam labeled data yang mayoritas.

3. Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi RapidMiner.
3. Modul Praktikum Data Warehousing dan Data Mining.

4. Langkah-langkah Praktikum

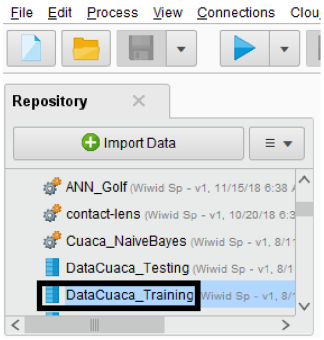
4.1. Menentukan Nilai k dan Tingkat Akurasi Algoritma K-NN

Pada praktikum ini kita akan melakukan sebuah prediksi seseorang akan bermain tennis atau tidak berdasarkan data training yang sudah ada dengan menggunakan algoritma K-Nearest Neighbour (K-NN). Sebelum melakukan prediksi terhadap data testing, langkah yang perlu dilakukan adalah menentukan jumlah k dengan tingkat akurasi yang paling tinggi.

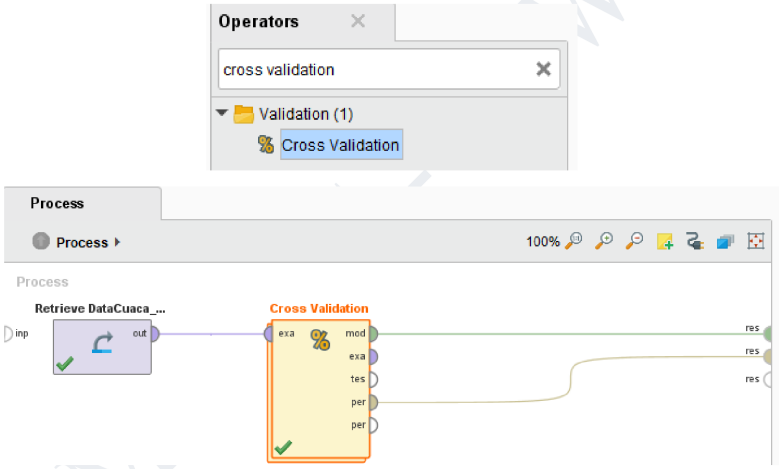
Berikut langkah-langkahnya:

1. Jalankan aplikasi RapidMiner.
2. Gunakan file data **DataCuaca_Training** pada repository sebagai data yang akan digunakan dalam proses penentuan jumlah k sesuai dengan tingkat akurasinya.

DATA MINING - WIMT SP

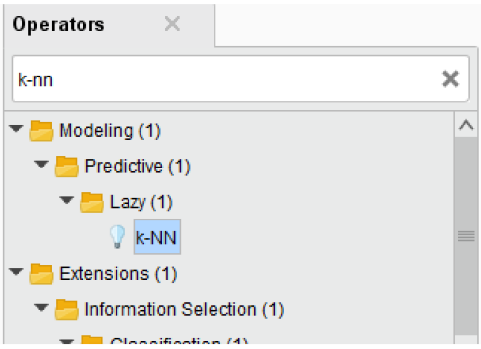


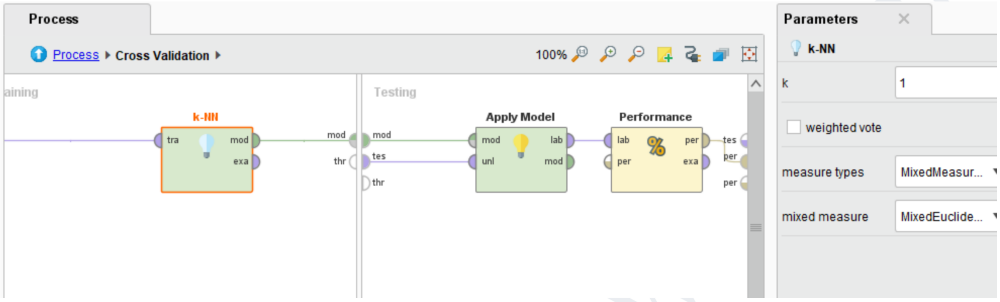
3. Drag **DataCuaca_Training** ke area Process View. Drag pula operator **Cross Validation** ke area Process View. Hubungkan port output data training ke port input Cross Validation serta port output Cross Validation dengan port input Result seperti gambar berikut.



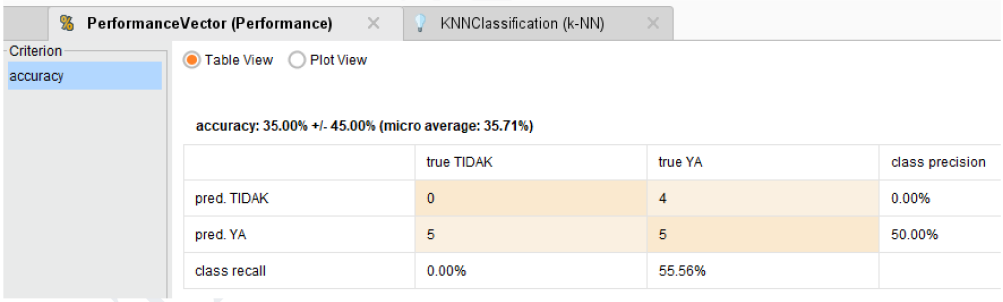
4. Selanjutnya adalah merancang algoritma k-nn. Klik ganda operator **Cross Validation** yang terdapat pada Process View, sehingga masuk ke jendela Process – Validation.
5. Masukkan operator **k-nn** dalam area Training, operator **Apply Model** dan **Performance** dalam area Testing.

   Klik pada operator k-NN dan tentukan nilai awal parameter k = 1 pada operator ini. Anda juga bisa memberikan tanda centang dalam pilihan **weighted vote** jika diperlukan untuk melihat tingkat akurasinya. Untuk saat ini biarkan kosong terlebih dahulu.



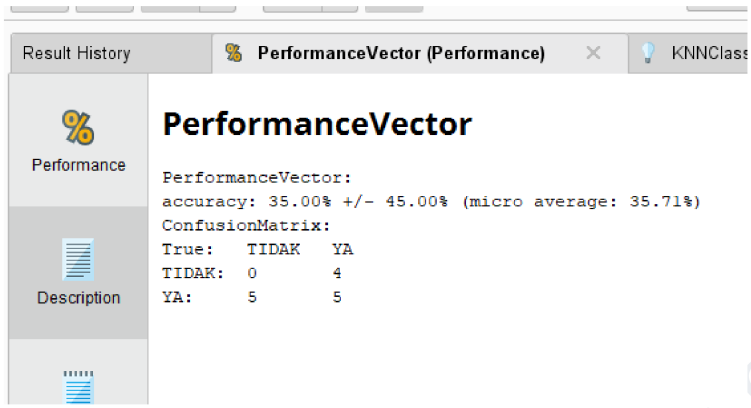


6. Jalankan proses dengan menekan tombol Run .
7. Berikut hasil proses klasifikasi dengan algoritma K-NN pada bagian PerformanceVector (Performance).

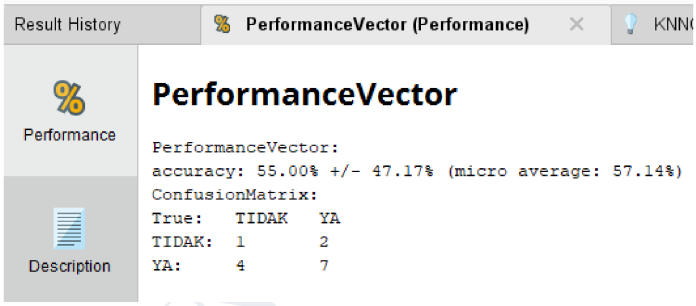


	true TIDAK	true YA	class precision
pred. TIDAK	0	4	0.00%
pred. YA	5	5	50.00%
class recall	0.00%	55.56%	

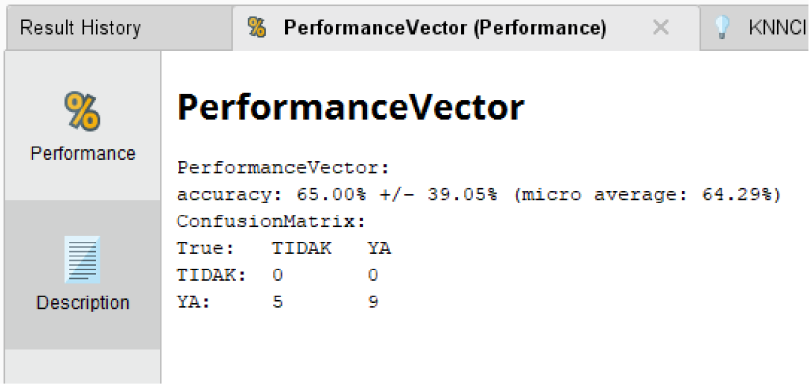
Dapat kita lihat bahwa tingkat accuracy pada percobaan ini mencapai 35%, yang menunjukkan angka keakuratan yang sangat rendah. Hasil ini juga bisa dilihat dalam mode Text View seperti berikut.



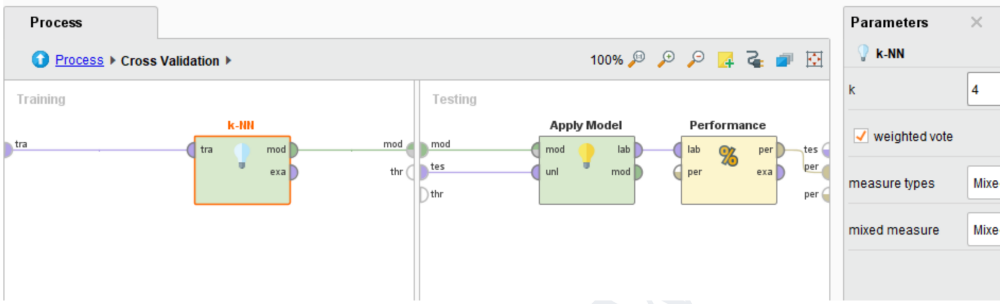
8. Dikarenakan hasil tingkat akurasi yang kurang, maka perlu dilakukan penambahan jumlah k pada operator k-NN. Ubahlah nilai parameter secara acak k = 6. Eksekusi kembali proses ini, dan lihatlah tingkat akurasi yang diperoleh sebesar 55% dengan mode Text View.

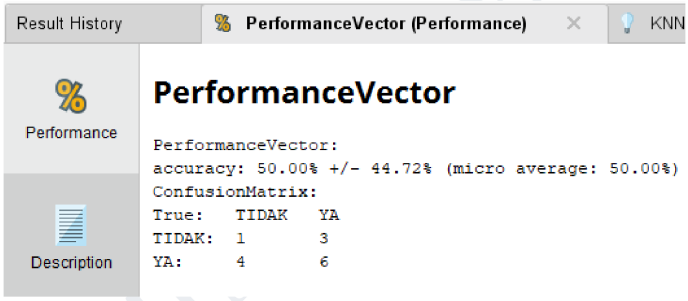


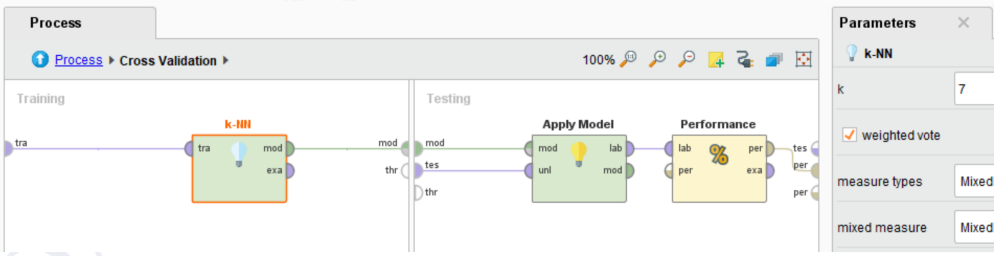
9. Ubahlah kembali nilai k dengan angka sembarang antara 1 sampai dengan 14 (14 merupakan jumlah data keseluruhan), jalankan kembali prosesnya. Apa yang terjadi? Ternyata tingkat akurasi maksimal yang diperoleh sebesar 65%, dan ini terjadi pada saat nilai k = 4, 7, 9, 11, 12, 13 dan 14.

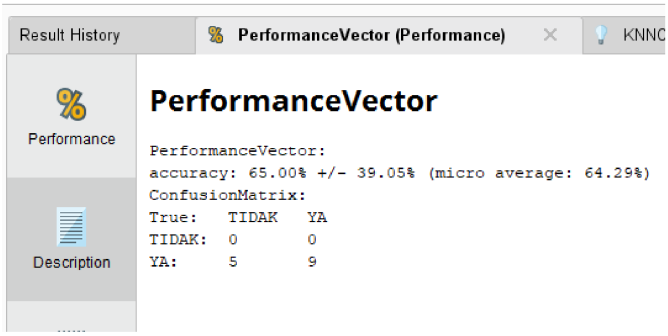


10. Pada operator k-NN, ubah kembali nilai k = 4 dan berikan tanda cek pada pilihan weighted vote. Kemudian jalankan kembali prosesnya dan lihatlah tingkat akurasinya. Ulangi langkah 10 ini dengan mengubah nilai k = 7. Bagaimana hasil tingkat akurasinya? Ternyata pada saat nilai k = 4 dengan menambahkan bobot weighted vote, tingkat akurasinya berkurang menjadi 50%, sedangkan ketika nilai k = 7, tingkat akurasinya tetap sama sebesar 65% meskipun diberi bobot weighted vote.









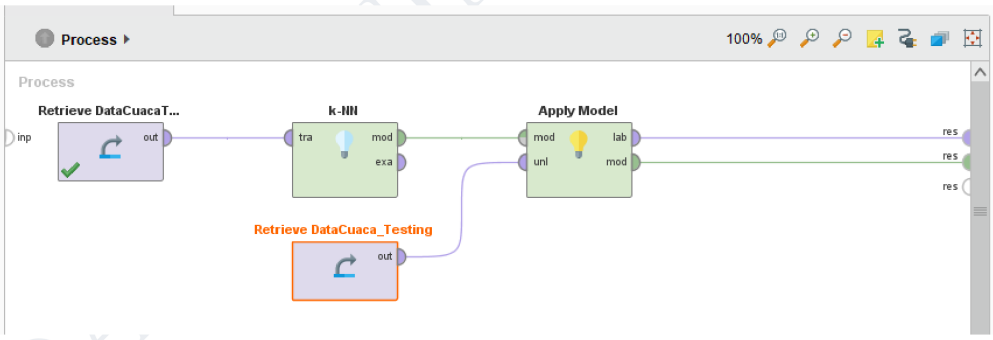
11. Sehingga dapat disimpulkan bahwa nilai k yang akan digunakan untuk melakukan prediksi terhadap data testing adalah k = 7.

4.2. Prediksi dengan Algoritma K-NN

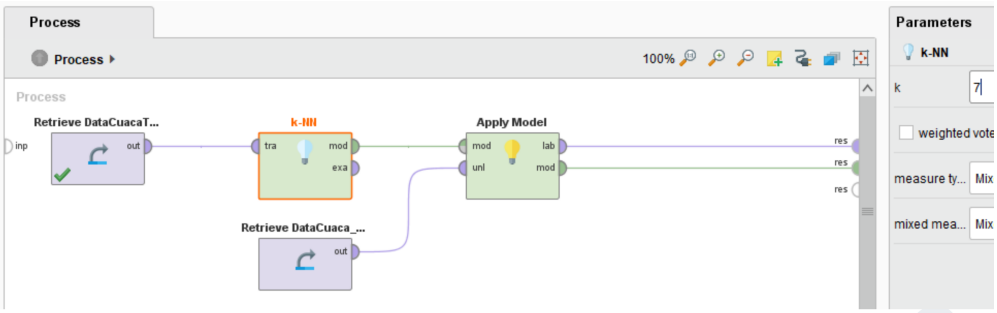
Setelah menentukan nilai k berdasarkan tingkat akurasinya, percobaan berikutnya adalah melakukan prediksi terhadap data testing yang sudah dipersiapkan.

Berikut langkah-langkah untuk melakukan prediksi terhadap data testing:

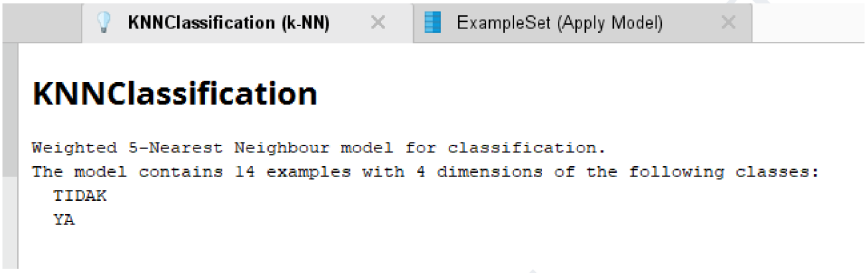
1. Masih menggunakan RapidMiner, pilih New Process.
2. Masukkan data training dan data testing. Gunakan file data **DataCuaca_Training** dan **DataCuaca_Testing** yang terletak pada repository. Masukkan ke dalam area Main Process.
3. Tambahkan operator **k-NN**. Hubungkan port out (output) dari data training ke port tra (training) pada operator k-NN.
4. Masukkan operator **Apply Model**. Hubungkan port output mod (model) dari operator k-NN dengan port input mod (model) pada operator Apply Model. Port out (output) dari data testing dihubungkan ke port unl (unlabelled data) pada operator Apply Model. Hubungkan pula port output lab (labelled data) dan mod (model) dari operator Apply Model ke panel res (result).



5. Klik pada operator k-NN, ubah nilai parameter k = 7 sesuai dengan tingkat akurasi yang diperoleh pada percobaan 4.1. Anda juga bisa memberikan bobot weighted vote pada proses ini meskipun tidak diharuskan.



6. Jalankan proses dengan menekan tombol Run .
7. Berikut hasil prosesnya:



ExampleSet (7 examples, 3 special attributes, 4 regular attributes) Filter (7 / 7 exam

Row No.	prediction(B...	confidence(...	confidence(...	Cuaca	Suhu	Kelembaban...	Berangin
1	YA	0.190	0.810	Cerah	75	65	TIDAK
2	YA	0.177	0.823	Cerah	80	68	YA
3	YA	0.442	0.558	Cerah	83	87	YA
4	TIDAK	0.569	0.431	Mendung	70	96	TIDAK
5	YA	0.184	0.816	Mendung	68	81	TIDAK
6	YA	0.218	0.782	Hujan	65	75	YA
7	YA	0.204	0.796	Hujan	64	85	YA

5. Tugas

1. Gunakan file DataStudi_Training pada repository sebagai data training dalam RapidMiner. Jika tidak ada buatlah tabel berikut sebagai data training. Simpan dengan nama DataStudi_Training dan masukkan pada repository. Variabel Lama_Studi sebagai dependent (bertipe label).

Jurusan_SMA	Gender	Asal_Sekolah	Rerata_SKS	Asisten	Lama_Studi
IPS	WANITA	SURAKARTA	18	TIDAK	TERLAMBAT
IPA	PRIA	SURAKARTA	19	YA	TEPAT
LAIN	PRIA	SURAKARTA	19	TIDAK	TERLAMBAT
IPA	PRIA	LUAR	17	TIDAK	TERLAMBAT
IPA	WANITA	SURAKARTA	17	TIDAK	TEPAT
IPA	WANITA	LUAR	18	YA	TEPAT
IPA	PRIA	SURAKARTA	18	TIDAK	TERLAMBAT
IPA	PRIA	SURAKARTA	19	TIDAK	TEPAT
IPS	PRIA	LUAR	18	TIDAK	TERLAMBAT
LAIN	WANITA	SURAKARTA	18	TIDAK	TEPAT
IPA	WANITA	SURAKARTA	19	TIDAK	TEPAT
IPS	PRIA	SURAKARTA	20	TIDAK	TEPAT
IPS	PRIA	SURAKARTA	19	TIDAK	TEPAT
IPA	PRIA	SURAKARTA	19	TIDAK	TEPAT
IPA	PRIA	LUAR	22	YA	TEPAT
LAIN	PRIA	SURAKARTA	16	TIDAK	TERLAMBAT
IPS	PRIA	LUAR	20	TIDAK	TEPAT
LAIN	PRIA	LUAR	23	YA	TEPAT
IPA	PRIA	SURAKARTA	21	YA	TEPAT
IPS	PRIA	SURAKARTA	19	TIDAK	TERLAMBAT

2. Carilah nilai k yang memiliki tingkat akurasi tertinggi dengan menggunakan algoritma k-NN dari data training tersebut!
3. Gunakan file DataStudi_Testing pada repository sebagai data testing. Atau buat baru dari tabel berikut sebagai data testing.

Jurusan_SMA	Gender	Asal_Sekolah	Rerata_SKS	Asisten
LAIN	WANITA	SURAKARTA	18	TIDAK
IPA	PRIA	SURAKARTA	19	YA
LAIN	PRIA	SURAKARTA	19	TIDAK
IPS	PRIA	LUAR	17	TIDAK
LAIN	WANITA	SURAKARTA	17	TIDAK
IPA	WANITA	LUAR	18	YA
IPA	PRIA	SURAKARTA	18	TIDAK
IPA	PRIA	SURAKARTA	19	TIDAK
IPS	PRIA	LUAR	18	TIDAK
LAIN	WANITA	SURAKARTA	18	TIDAK

4. Lakukan prediksi terhadap data testing tersebut menggunakan algoritma k-NN. Tuliskan tabel hasil prediksinya!